CS464 Introduction to Machine Learning
Spring 2020
Homework 2

Due: April 26, 2020 11:59 PM

Instructions

- For this homework, you may code in any programming language of your choice.
- You are NOT allowed to use any machine learning packages, libraries or toolboxes for this assignment (such as scikit-learn, tensorflow, keras, theano, MATLAB Statistics and Machine Learning Toolbox functions, e1071, nnet, kernlab etc.) unless otherwise stated.
- Submit a soft copy of your homework to Moodle.
- Upload your code and written answers to the related assignment section on Moodle (.TAR or .ZIP). Submitting hard copy, handwritten or scanned files is NOT allowed.
- The name of your compressed folder must be "`CS464_HW2_Section#_Firstname_Lastname`" (i.e., `CS464_HW2_1_sheldon_cooper`). Please do not use any Turkish characters in your compressed folder name.
- Your code should be in a format that is easy to run and must include a driver script serving as an entry point. You must also provide a README file with clear instructions on how to execute your program.
- This is an individual assignment for each student. That is, you are NOT allowed to share your work with your classmates.
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.
- If you have any questions, email to `doruk.cakmakci@bilkent.edu.tr`.

1 SVM and Logistic Regression

In this question, you are expected to classify aquatic animals using features extracted with two different methods: transfer learning with Inception v3 and histogram of oriented gradients (HOG). Transfer learning is a widely used approach for various tasks such as object recognition and localization. Inception v3 is a deep convolutional neural network architecture developed by Google [1]. In order to extract features from aquatic animal images, an Inception v3 model is pre-trained on ImageNet [2] dataset. Then the model's last layer is removed to use the output of second to last layer (of size 2048×1) as image features. Finally, each image in the dataset is passed from the model for feature extraction. On the other hand, HOG [3] is a feature descriptor used in computer vision for the purpose of object detection, which counts occurrences of gradient orientations in different parts of the image. HOG produces features of size 324×1. Image features are extracted and provided for your use.

Aquatic animals dataset is a subset of CIFAR100 dataset [4]. Image features and labels are provided in `q1_dataset.mat`. The dataset contains a train split of 1000 images and a test split of 200 images. You will not use a separate validation split. Each image in the dataset has a subclass label and the corresponding superclass label. Beaver (Subclass 0), dolphin (Subclass 1), otter (Subclass 2), seal (Subclass 3) and whale (Subclass 4) classes are subclasses of aquatic mammal (Superclass 0). In like manner, aquarium fish (Subclass 0), flatfish (Subclass 1), ray (Subclass 2), shark (Subclass 3) and trout (Subclass 4) are subclasses of fish superclass (Superclass 1). `q1_dataset.mat` consists of the following:

- **inception_features_train:** A matrix of size 2000×2048 whose rows correspond to the features extracted from training split using Inception v3 network.
- **inception_features_test:** A matrix of size 400×2048 whose rows correspond to the features extracted from test split using Inception v3 network.
- **hog_features_train:** A matrix of size 2000×324 whose rows correspond to the features extracted from training split using HOG method.
- **hog_features_test:** A matrix of size 400×324 whose rows correspond to the features extracted from training split using HOG method.
- **superclass_labels_train:** A matrix of size 2000×1 which contains the superclass label assignments for train split.
- **superclass_labels_test:** A matrix of size 400×1 which contains the superclass label assignments for test split.
- **subclass_labels_train:** A matrix of size 2000×1 which contains the subclass label assignments for train split.
- **subclass_labels_test:** A matrix of size 400×1 which contains the subclass label assignments for test split.

You are expected to measure *training times* of all models.

Superclass Classification Using Logistic Regression

For this part you will use the neural network generated features and HOG features to predict the *superclass* of an image. You will train each algorithm on HOG features and on neural network generated features separately. You should not test a model trained on HOG features with neural network generated features (vice versa). You must present two sets of results for each part. Assume that fish is the positive class.

Question 1.1 **[12 pts]** Implement mini-batch gradient ascent algorithm with *batch size = 25* and stochastic gradient ascent algorithm to train logistic regression models. Initialize all weights to random numbers drawn from a Gaussian distribution $N(0, 0.01)$. Try different learning rates and choose the one which works best for you. Use 1000 iterations to train your model. Report accuracy, precision, recall, negative predictive value (NPV), false positive rate (FPR), false discovery rate (FDR), F1, F2 scores and the confusion matrix using your model on the corresponding test set. Discuss your results.

Question 1.2 **[8 pts]** Implement full batch gradient ascent algorithm to train your logistic regression model. Initialize all weights to random numbers drawn from a Gaussian distribution $N(0, 0.01)$. Use the learning rate you have chosen in Question 1.1 and perform 1000 iterations to train your models. Print model weights in each iteration $i \in \{100, 200, ..., 1000\}$. Report 10 most important features(in terms of indices) and discuss the relation between weights and individual importance of features. Report accuracy, precision, recall, negative predictive value (NPV), false positive rate (FPR), false discovery rate (FDR), F1, F2 scores and the confusion matrix using your model on the given test set. Discuss your results.

Superclass Classification Using SVM

For this part you will use the neural network generated features and HOG features to predict the *superclass* of an image. You will train each algorithm on HOG features and on neural network generated features separately. You should not test a model trained on HOG features with neural network generated features (vice versa). You must present two sets of results for each part. Assume fish superclass is the positive class.

Question 1.3 **[0 pts]** Implement `stratified_k_fold(features, labels, k)` function which shuffles and splits a dataset into k folds with approximately equal label ratios.

Question 1.4 **[6 pts]** Train a soft margin SVM model with linear kernel. Tune C hyper-parameter of the model using 5-fold cross validation on the training dataset. Use the function implemented in Question 1.3 to perform cross validation. Calculate a performance metric of your choice for each left-out fold. Report mean of the selected performance metric calculated on the left-out fold for $C \in \{10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}\}$. You should select the optimal value of C, denoted by C^*, based on mean of the selected performance metric. Train a soft margin SVM model with linear kernel where $C = C^*$ on training dataset and test your model on the corresponding test dataset. Report accuracy, confusion matrix, precision and recall. You should perform all steps for both HOG features and neural network generated features, and compare their results.

Question 1.5 **[6 pts]** Train a hard margin SVM with radial basis function (rbf) kernel. Radial basis function is defined as:

$$K(x, x') = exp(-\gamma||x - x'||^2) \tag{1.1}$$

where γ is an hyper-parameter of the model which determines the radius of influence of support vectors on decision boundary of the model. Tune γ of the model using 5-fold cross validation on the training dataset. Use the function implemented in Question 1.3 to perform cross validation. Calculate a performance metric of your choice for each left-out fold. Report mean of the selected performance metric calculated on the left-out fold for $\gamma \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{6}\}$. You should select the optimal value of γ, denoted by γ^*, based on mean of the selected performance metric.
Train a hard margin SVM with rbf kernel where $\gamma = \gamma^*$ on training dataset and test your model on the corresponding test dataset. Report accuracy, confusion matrix, precision and recall. You should perform all steps for both HOG features and neural network generated features, and compare their results.

Question 1.6 **[6 pts]** Train a soft margin SVM with radial basis function (rbf) as kernel. Tune the hyper-parameters of the model using 5-fold cross validation on the training dataset. Use the function implemented in Question 1.3 to perform cross validation. Calculate a performance metric of your choice for each left-out fold. Report mean of the selected performance metric calculated on the left-out fold for $(C, \gamma) \in \{10^{-2}, 1, 10^{2}\} \times \{2^{-2}, 2^{1}, 2^{6}\}$. You should select the optimal values of C and γ, denoted by C^* and γ^*, based on mean of the selected performance metric.
Train a soft margin SVM with rbf kernel where $C = C^*$ and $\gamma = \gamma^*$ on training dataset and test your model on the corresponding test dataset. Report accuracy, confusion matrix, precision and recall. You should perform all steps for both HOG features and neural network generated features, and compare their results.

Subclass Classification Using SVM

For this part you will use the neural network generated features and HOG features to predict the *subclass* of an image. You will train each algorithm on HOG features and on neural network generated features separately. You should not test a model trained on HOG features with neural network generated features (vice versa). You must present two sets of results for each part.

Question 1.7 **[6 pts]** Train a soft margin SVM with radial basis function(rbf) as kernel and one-vs-all approach. Tune the hyper-parameters of the model using 5-fold cross validation on the training dataset. Use

the function implemented in Question 1.3 to perform cross validation. Calculate a performance metric of your choice for each left-out fold. Report mean of the selected performance metric calculated on the left-out fold for $(C, \gamma) \in \{10^{-2}, 1, 10^{2}\} \times \{2^{-2}, 2^{1}, 2^{6}\}$. You should select the optimal values of C and γ, denoted by C^* and γ^*, based on mean of the selected performance metric.
Train a soft margin SVM with rbf kernel where $C = C^*$ and $\gamma = \gamma^*$ on training dataset and test your model on the corresponding test dataset. Report class-based accuracy, micro and macro averages of precision and recall, confusion matrix. You should perform all steps for both HOG features and neural network generated features, and compare their results.

Question 1.8 [6 pts] Train a hard margin SVM with polynomial kernel. Tune the hyper-parameters of the model (degree of the polynomial and γ) using 5-fold cross validation on the training dataset. Use the function implemented in Question 1.3 to perform cross validation. Calculate a performance metric of your choice for each left-out fold. Report mean of the selected performance metric calculated on the left-out fold for $(d, \gamma) \in \{3, 5, 7\} \times \{2^{-2}, 2^{1}, 2^{6}\}$. You should select the optimal values of d and γ, denoted by d^* and γ^*, based on mean of the selected performance metric.
Train a hard margin SVM with polynomial kernel where $d = d^*$ and $\gamma = \gamma^*$ on training dataset and test your model on the corresponding test dataset. Report class-based accuracy, confusion matrix, micro and macro averages of precision, recall and F1 score. You should perform all steps for both HOG features and neural network generated features, and compare their results.

Comparison of Models

Question 1.9 [12 pts] Compare the performance of models constructed in previous sections. In particular, elaborate on following items for each task:

- Which feature extraction method yields better results.
- Which model and feature combination performed best/worst.

Discuss following items in detail for each feature extraction method:

- The effect of C on the decision boundary of SVM.
- The effect of γ on the decision boundary of SVM with RBF kernel.
- The effect of d on the decision boundary of SVM with polynomial kernel.
- The effects of different (C, γ) pairs on the decision boundary and tolerance of SVM.
- The effects of different (d, γ) pairs on the decision boundary of SVM with polynomial kernel.
- The effect of batch size to performance and training time of logistic regression models.

Also discuss the use of γ parameter for polynomial kernel. What are the advantages and disadvantages of logistic regression and SVM models compared to each other?

Question 1.10 [8 pts] In what cases NPV, FPR, FDR and F1 score would be more informative compared to accuracy, precision, recall alone? Discuss if accuracy is a reliable metric for subclass and superclass classification tasks.

2 PCA

In this question, you will compare SVD based and covariance matrix based implementations of PCA. You are not allowed to use any PCA implementations for this question. You will apply PCA on `q2_dataset.mat` which consists of 150 leaf images. The dataset is a subset of [5]. Each image is a binary image (i.e. pixel values are either 1 or 0) of size 85×125. The dataset is provided to you as a tensor of size $150 \times 85 \times 125$. Lets denote the dataset tensor as X. First, flatten each image by reshaping X to a matrix of size 150×10625.

Question 2.1 **[8 pts]** Apply SVD based implementation of PCA to X. Report the runtime of the algorithm. Reconstruct all images in the dataset using all principal components. Calculate and report MSE between the original and reconstructed version of the dataset. Discuss your results.

Question 2.2 **[8 pts]** Apply covariance matrix based implementation of PCA to X. Report the runtime of the algorithm. Reconstruct all images in the dataset using all principal components. Calculate and report MSE between the original and reconstructed version of the dataset. Discuss your results. (Hint: Discard the imaginary parts of reconstructed images).

Question 2.3 **[8 pts]** Plot the following to a grid of size 3×5: (i) First 5 images in the dataset, and (ii) reconstructed versions of first 5 images in the dataset from Question 2.1 and Question 2.2. Compare the implementations in terms of runtime performance, precision of reconstruction and numerical stability. Is MSE a reliable metric for comparing image similarity? Can reliability of MSE metric be improved for measuring image similarity? Discuss.

Question 2.4 **[6 pts]** Assume that you are going to compare running time of PCA implementations on another dataset, X_{new}, of size $n_{samples} \times n_{features}$ where each feature is real-valued. Order the expected running time of the implementations when (i) $n_{samples} >> n_{features}$, (ii) $n_{samples} \sim n_{features}$, and (iii) $n_{samples} << n_{features}$. Discuss your claims.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[4] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[5] "Leaf classification challenge," https://www.kaggle.com/c/leaf-classification/data.